

第二次课堂练习

一. 名词解释

1. 地址映射

答: 地址映射是计算机系统中将一种地址转换为另一种地址的过程。它的作用是使程序能够使用逻辑地址。

2. 动态重定位

答: 动态重定位是通过硬件在程序运行时将逻辑地址转换为物理地址, 使程序可以装载到任意内存位置并安全运行的机制。

3. 虚拟存储器

答: 具有请求调入功能和置换功能, 能从逻辑上对内存容量加以扩充的一种存储器系统。

4. 静态链接

答: 静态链接是指程序编译或连接阶段, 将所需的目標文件和库文件中的代码全部链接到一个可执行文件中的过程。

5. 对换

答: 把内存中暂时不能运行的进程或者暂时不用的程序和数据, 调出到外存上, 以便腾出足够的内存空间, 再把已具备运行条件的进程或进程所需的程序或数据, 调入内存。

6. 设备驱动程序

答: 设备驱动程序是操作系统 I/O 系统中负责连接上层软件与设备控制器的通信程序。它的主要作用是将上层发出的抽象 I/O 请求转换为具体的硬件操作命令, 并将设备控制器返回的信号反馈给上层软件。

7. Spooling

答: 外围操作与 CPU 对数据的处理同时进行, 这种在联机情况下实现的同时外围操作称为 Spooling。

8. I/O 通道

答: 使一些原来由 CPU 处理的 I/O 任务转由通道来承担, 从而把 CPU 从繁杂的 I/O 任务中解脱出来。

9. 文件系统

答: 文件系统是 OS 的一部分, 它提供了一种管理机制, 以便 OS 对自身及所有用户的数据与程序进行在线存储和访问。

10. 目标文件

答: 指由“把源程序经过编译程序编译后, 但尚未经过链接程序链接的目标代码”所构成的文件, 其后缀是“.obj”。

11. 文件的逻辑结构

答: 指从用户角度出发所观察到的文件组织形式, 即文件是由一系列的逻辑记录所组成的, 是用户可以直接处理的数据结构, 它独立于文件的物理特性。
及其

12. 有结构文件

答: 在记录式文件中, 每个记录都用于描述实体集中的一个实体, 各记录有着相同或不同数目的数据项。

13. 位示图

答: 利用二进制的位来表示磁盘中一个盘块的使用情况, 磁盘上的所有盘块都有一个二进制位与之对应, 这样, 由所有盘块所对应的位构成的一个集合, 称为位示图。

14. 程序接口

答: 该接口是为用户程序在执行中访问系统资源而设置的, 它是由一组系统调用组成的, 每个系统调用都是一个能完成特定功能的子程序。

15. 系统调用

答: 应用程序请求OS内核完成某功能时的一种过程调用; 用户与内核的接口。

16. I/O中断

答: 指CPU对I/O设备发来的中断信号的一种响应。

17. 文件管理系统

答: 文件管理系统是操作系统中用于实现对文件的创建存储, 访问, 修改删除等操作的模块, 通过将其管理的程序和数据组成一系列文件。

18. 文件

答: 指具有文件名的若干相关元素的集合。

19. 文件的逻辑结构

答: 与11相同。

20. 文件的物理结构

答: 指系统将文件存储在外部上所形成的一种存储组织形式, 是用户所看不见的。

二. 填空题

1. 静态重定位 2. 绝对装入式, 静态重定位装入式, 动态重定位装入式
3. 静态链接式, 装入时动态链接, 运行时动态链接
4. 空闲分区链, 空闲分区表 5. 紧凑或拼接 6. 固定, 系统, 不固定, 用户
7. 对换性, 虚拟性 8. 文件管理有关的软件, 被管理的文件, 实施文件管理所需的数据结构
9. 分区的分配方式, 分区的回收方式, 碎片的处理方式
10. 内存碎片 11. 分页存储管理, 分段存储管理, 段页式存储管理
12. 快表 13. 逻辑地址空间, 物理内存空间 14. 方便编程, 信息共享, 信息保护, 动态链接, 动态增长
15. 内存管理单元
16. 局部性, 时间局部性, 空间局部性 17. 处理机, 指令类型单一, 没有自己的内存
18. 顺序文件, 索引文件, 索引顺序文件 19. 多级索引组织方式, 增量式索引组织方式, 混合索引方式
20. 动态重定位 21. 中断处理程序, 设备驱动程序, 设备独立性软件
22. 顺序文件, 链接文件, 索引文件, 索引文件
23. 高速缓冲存储器, 主存储器, 辅存储器.
24. 连续分配方式, 离散分配方式 25. 外存的分派方式, 存储介质的特性
26. 顺序文件, 索引文件, 索引顺序文件
27. 连续分配, 链接分配, 索引分配
28. 绝对路径, 相对路径.
29. 空闲盘块链, 空闲盘区链
30. 硬链接, 软链接
31. 块设备, 字符设备, 独占设备, 共享设备, 虚拟设备
32. DMA方式, 通道方式

33. 单缓冲, 环形缓冲, 缓冲池

34. 寻道时间, 存取时间

三. 简答题

1. (1) 计算机存储系统主要有寄存器, 高速缓存, 主存储器, 辅存储器, 外部存储器。

(2) 寄存器位于CPU内部, 如程序计数器(PC)、累加器、通用寄存器等

高速缓存有 L1(一级)、L2(二级)、L3(三级) 缓存

主存有动态随机访问内存 DRAM

辅存储器有硬盘、固态硬盘

外部存储器有 U盘、光盘、磁带等

(3) 从存储性能看, 计算机采用分层结构在速度、容量和成本间取得平衡。越靠近CPU, 速度越快, 容量越小, 成本高; 越远则容量大但速度慢。通过局部性原理, 将常用数据缓存在上层, 并结合多级缓存、虚拟内存等技术, 提升系统效率与经济性。

2. 计算机程序从代码编写到运行完毕所经历的主要过程有:

1. 编写源代码, 2. 编译 3. 链接 4. 加载 5. 运行执行 6. 结束退出

3. 程序的链接主要有静态链接, 装入时动态链接, 运行时动态链接。

静态链接对相对地址进行修改, 变换外部调用符号;

装入时动态链接便于修改和更新, 便于实现对目标模块的共享

运行时动态链接加快装入过程, 节省大量的内存空间

4. (1) 内存动态分区分配算法根据搜索空闲区的方式可分为基于顺序搜索的动态分区分配算法 (顺序分配算法)、基于索引搜索的动态分区分配算法 (索引分配法)。

(2) 基于顺序搜索的动态分区分配算法 (顺序分配法) 的典型算法有:

① 首次适应算法, 核心思想是从空闲分区链的开头开始, 按地址递增的顺序查找第一个能够满足作业所需大小的空闲分区, 并将其分配给该作业, 剩余部分继续保留在链中

② 循环首次适应算法, 核心思想是从上次分配结束的位置开始查找, 按地址顺序查找第一个满足要求的空闲分区, 查找成循环进行, 避免总是占用低地址空间。

③ 最佳适应算法, 核心思想是在所有能满足要求的空闲分区中, 选择最小但足够使用的分区进行分配, 以尽量减少浪费。

④ 最坏适应算法, 核心思想是总是选择最大的空闲分区进行分配, 保留较大的剩余空间, 以减少产生小碎片的可能。

(3) 基于索引搜索的动态分区分配算法(索引分配算法)的典型算法有:

① 快速适应算法, 核心思想是将空闲分区按常用大小分类管理, 通过索引表快速定位合适的空闲链表, 直接分配整块分区, 提高查找效率, 属于以空间换时间的策略。

② 伙伴系统, 核心思想是将内存分区大小限定为2的幂次方, 根据需要分裂或合并为“伙伴”分区, 便于快速分配和高效回收, 支持动态合并释放空间。

③ 哈希算法, 核心思想是通过哈希函数以空闲分区大小为关键字建立哈希表, 快速定位对应的空闲分区链表, 实现高效的查找与分配。

5. 内存动态分区分配算法有: 首次适应算法, 循环首次适应算法, 最佳适应算法, 最坏适应算法, 快速适应算法, 伙伴系统, 哈希算法。

CPU调度算法有: 先来先服务(FIFS), 最短作业优先(SJF), 时间片轮转(RR), 优先级调度, 多级反馈队列。

内存分区算法负责分配内存空间, CPU调度算法分配处理时间, 二者分配资源不同, 但都是为了提高系统效率, 实现资源的合理利用与多道程序并发执行。

6. 页面置换算法有: 最佳页面置换算法, 先进先出页面置换算法, 最近最久未使用页面置换算法, 最少使用页面置换算法, clock页面置换算法, 页面缓冲算法。

页面置换算法用于内存管理, 决定淘汰哪一页; CPU调度算法用于进程管理, 决定哪个进程执行。二者调度对象不同, 但目的相同, 都是优化资源利用和系统性能。

7. 列举典型的页面置换算法和磁盘调度算法，并分析两者之间的异同点。

典型的页面置换算法有：最佳页面置换算法、先进先出页面置换算法、最近最久未使用页面置换算法、最少使用页面置换算法、Clock 页面置换算法、页面缓冲算法。

典型的磁盘调度算法有：FCFS 调度算法、SSTF 调度算法、SCAN 调度算法、CSCAN 调度算法、NStepSCAN 调度算法和 FSCAN 调度算法。

两者之间的异同点是：页面置换算法用于内存管理，负责在页面不足时选择淘汰页，优化内存使用；磁盘调度算法用于磁盘管理，决定磁头访问顺序，优化磁盘访问时间。两者调度对象不同，一个管理内存页，另一个管理磁道请求，但本质上都是资源调度策略，目标都是提高系统性能与资源利用率。

8. 分别简述分页存储、分段存储和段页存储的地址变换过程。

分页存储的地址变换过程：分页将逻辑地址空间划分为固定大小的页，物理内存划分为等大小的页框。逻辑地址由页号 (Page Number) 以及页内偏移 (Offset) 组成；操作系统通过页表将页号转换为物理页框号；最终的物理地址 = 页框号 \times 页大小 + 页内偏移。

分段存储的地址变换过程：分段将逻辑地址划分为若干功能段，如代码段、数据段、栈段等，每段大小不固定。逻辑地址由段号 (Segment Number) 以及段内偏移 (Offset) 组成；操作系统通过段表获取段的起始物理地址 (基址)；检查偏移是否越界，最终物理地址 = 段基址 + 段内偏移。

段页式存储的地址变换过程：段页式是分页与分段的结合，每个段再划分为若干页，既支持结构化又简化内存分配。逻辑地址由段号、页号、页内偏移组成；操作系统先通过段号查找段表，获取页表地址，再通过页号查页表，得到物理页框号，最终物理地址 = 页框号 \times 页大小 + 页内偏移。

9. 简述中断处理程序的各个处理步骤。

中断处理程序在系统接收到中断信号后，会按以下步骤进行处理：

保存现场：保存当前正在运行程序的状态（如程序计数器、寄存器等），以便中断处理结束后能继续执行。

中断响应：CPU 识别中断信号，暂停当前程序执行，通过中断向量表查找对应中断服务程序入口地址。

转向中断服务程序 (ISP)：根据中断类型跳转执行对应的中断处理程序，处理外设请求、错误等事件。

处理中断请求：执行具体的中断服务逻辑，如读取数据、完成设备控制等。

恢复现场：中断服务程序执行完毕后，恢复先前保存的程序状态。

返回原程序：使用中断返回指令（如 IRET）跳回中断发生前的程序，继续执行。

10. 从用途、数据类型、组织和管理方式等角度简述文件类型分类。

按用途分类：文件可分为系统文件（如操作系统核心文件）、用户文件（如源代码、数据文件）和库文件（如标准函数库），系统文件主要由操作系统控制，

用户和库文件可被调用使用。

按数据类型分类：分为源文件（由源程序和输入数据组成）、目标文件（编译后但未链接的中间文件，如 .obj）和可执行文件（编译链接完成后可直接运行的文件，如 .exe）。

按访问权限分类：包括可执行文件（仅允许执行）、只读文件（仅允许读取）和读/写文件（允许读取和修改），体现系统对文件访问控制的管理。

按组织与处理方式分类：可分为普通文件（如文本、程序和数据文件）、目录文件（用于存储目录信息，支持文件检索和组织）和特殊文件（主要指设备文件，操作方式与普通文件一致，由设备驱动程序控制）。

11. 请详述文件目录的分类及相应查询方式。

文件目录的分类包括单级目录、两级目录、树形目录和无环图目录，每种结构对文件管理的灵活性与查询效率有不同影响：

单级目录结构最简单，所有文件在同一目录下，按名唯一但查找效率低，不能支持重名和用户隔离，适合单用户场景；

两级目录为每个用户建立独立目录，支持用户隔离与同名文件，但不利于跨用户共享；

树形目录通过分级结构组织文件与子目录，支持绝对路径与相对路径访问，便于管理和权限控制，是现代操作系统广泛使用的目录结构；

无环图目录允许一个文件或子目录被多个父目录引用，支持对称共享，是树形结构的扩展，避免了共享局限性。

查询方式主要有两种：

线性检索法：依次对目录项逐级顺序查找，适合路径结构明确或使用通配符的情况；

哈希法：将文件名转换为索引值直接定位，提高检索效率，但需解决哈希冲突。

12. 请简述文件结构的三种主要组织方式，并对比分析各自优劣。

文件的三种主要组织方式包括：顺序结构、索引结构和散列结构。

顺序结构优点是存储简单，管理方便，并且顺序访问效率高，适合批量处理。其缺点是插入、删除不便，需重新排列或重建文件，不适合频繁修改或随机访问场景。

索引结构优点是随机访问性能高，定位快速，并且插入和删除操作效率较顺序结构高。支持范围查询和多种排序方式。其缺点是占用额外索引空间，索引维护复杂，尤其在频繁更新时性能下降。

散列结构优点是查找速度快，适合等值查询（如查找某 ID），插入效率高。其缺点是不支持顺序访问和范围查询，哈希冲突可能影响效率，处理较复杂。文件扩展困难，负载高时性能下降明显。

13. 请分别简述内存和外存的存储分配空间分配方式，并对比分析它们之间的异同点。

内存的空间分配方式有连续分配方式和非连续分配方式，其中连续分配方式包括：

单一连续分配：所有内存空间为操作系统和单一用户进程所用。

固定分区分配：内存被划分为若干固定大小的分区，进程按其大小分配到适合的分区。

可变分区分配（动态分区）：根据进程需要动态划分内存，常结合分区分配算法如首次适应、最佳适应、循环适应等。

非连续分配方式包括：

分页（Paging）：将内存和进程划分为固定大小的页（Page）和页框（Frame），实现离散分配。

分段（Segmentation）：根据程序逻辑结构划分段，段大小不一，分配内存段。

段页式管理：综合分页和分段技术，既支持逻辑结构，又便于物理分配。

外存的空间分配方式有连续分配方式和多级索引、混合分配。其中连续分配方式包括：

连续分配：将整个文件放在一段连续磁盘块中，查找快但易产生外部碎片。

链接分配：文件分散存放，每个块指向下一个，适用于顺序访问。

索引分配：使用索引块存储所有数据块号，适合随机访问。

多级索引、混合分配有如 UNIX 的 i-node 结构，采用直接索引、间接索引和多级索引结合，可支持大文件存储。

14. 请分别简述提高磁盘 I/O 速度的多种途径。

提高磁盘 I/O 速度的途径主要包括以下几种方式：

使用磁盘高速缓存；

优化磁盘调度算法；

采用预读技术；

使用磁盘阵列技术；

改进文件的目录结构与检索方式；

改进文件的物理存储结构；

使用固态硬盘替代传统机械硬盘。